JAVA.UTIL.ARRAYLIST CLASS

Introduction

The **java.util.ArrayList** class provides resizable-array and implements the **List** interface.Following are the important points about ArrayList:

- It implements all optional list operations and it also permits all elements, includes null.

- It provides methods to manipulate the size of the array that is used internally to store the list.

- The constant factor is low compared to that for the LinkedList implementation.

Class declaration

Following is the declaration for **java.util.ArrayList** class:

```
public class ArrayList<E>
    extends AbstractList<E>
        implements List<E>, RandomAccess, Cloneable, Serializable
```

Here **<E>** represents an Element. For example, if you're building an array list of Integers then you'd initialize it as

```
ArrayList<Integer> list = new ArrayList<Integer>();
```

Class constructors

S.N.	Constructor & Description
1	**ArrayList()** This constructor is used to create an empty list with an initial capacity sufficient to hold 10 elements.
2	**ArrayList(Collection<? extends E> c)** This constructor is used to create a list containing the elements of the specified collection.
3	**ArrayList(int initialCapacity)** This constructor is used to create an empty list with an initial capacity.

Class methods

S.N.	Method & Description
1	boolean add(E e) This method appends the specified element to the end of this list.
2	void add(int index, E element) This method inserts the specified element at the specified position in this list.
3	boolean addAll(Collection<? extends E> c)

	This method appends all of the elements in the specified collection to the end of this list, in the order that they are returned by the specified collection's Iterator
4	**boolean addAll(int index, Collection<? extends E> c)** This method inserts all of the elements in the specified collection into this list, starting at the specified position.
5	**void clear()** This method removes all of the elements from this list.
6	**Object clone()** This method returns a shallow copy of this ArrayList instance.
7	**boolean contains(Object o)** This method returns true if this list contains the specified element.
8	**void ensureCapacity(int minCapacity)** This increases the capacity of this ArrayList.
9	**E get(int index)** This method returns the element at the specified position in this list.
10	**int indexOf(Object o)** This method returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element.
11	**boolean isEmpty()** This method returns true if this list contains no elements.
12	**int lastIndexOf(Object o)** This method returns the index of the last occurrence of the specified element in this list, or -1 if this list does not contain the element.
13	**E remove(int index)** This method removes the element at the specified position in this list.
14	**boolean remove(Object o)** This method removes the first occurrence of the specified element from this list, if it is present.
15	**protected void removeRange(int fromIndex, int toIndex)** This method removes from this list all of the elements whose index is between fromIndex(inclusive) and toIndex(exclusive).
16	**E set(int index, E element)** This method replaces the element at the specified position in this list with the specified element.
17	**int size()** This method returns the number of elements in this list.
18	**Object[] toArray()** This method returns an array containing all of the elements in this list in proper sequence (from first to last element).
19	**<T> T[] toArray(T[] a)** This method returns an array containing all of the elements in this list in proper sequence (from first to last element); the runtime type of the returned array is that of the specified array.

20	void trimToSize()
	This method trims the capacity of this ArrayList instance to be the list's current size.

Methods inherited

This class inherits methods from the following classes:

- java.util.AbstractList

- java.lang.AbstractCollection

- java.util.Object

- java.util.List